



Release Note ZFRN-009

*OEMmodule™ 486
BIOS version E*

ZF MicroSystems, Incorporated ■ 1052 Elwell Court, Palo Alto, CA 94303 ■ Tel: 650 965-3800 ■ Fax: 650 965-4050

Summary

This application note describes the features of the E version of the *OEMmodule 486* BIOS. ZF will continue to develop and enhance the BIOS, and will provide updates to all *OEMmodule 486* customers as the newer versions are released.

Information contained in this document:

- DiskOnChip2000
- DOS management
- BIOS Features
- Release Media: Disk Images/Files.
- Flash Architecture
- Installation process
- 486 Post Code Listing

DiskOnChip2000

NOTE: It is important that the DiskOnChip firmware update be performed before the BIOS update.

This release includes an updated version of DiskOnChip2000 firmware. The newer version fixes many bugs present in the older (*OEMmodule 386*) version. It also now supports Real Time Operating Systems such as QNX. In order to update your DiskonChip firmware the following command must be typed at the DOS prompt: **DUPDATE /WIN:D800 /S:DOC110.EXB /FIRST**. The DiskonChip utilities disk must be in the drive.

DOS Management

The size of the flash disk currently allows DOS to be loaded on the system as part of a SYS process common to all diskette or hard drives. After formatting the drive, the command **SYS** "flash drive letter" will transfer the system files from the diskette to the RFD. As an example, if the source of the operating system is a diskette in the A drive and you have defined the internal flash as the B drive in the CMOS setup screen, the command **SYS B:** typed at the A: prompt will transfer the files to the RFD.

BIOS

Improvements

Fast Page DRAMs – the BIOS now supports Fast Page DRAM auto detection.

Fixed Timer initialization – Real time clock lost 4 minutes per day.

Custom CMOS Defaults – some of the custom CMOS defaults were changed:

- Byte wide socket is configured as a generic ROM & 16K.

- System speed → 100 MHz
- Super IO registers → UNLOCKED
- A20 after Post → DISABLED.

Features

The version E release supports the following features:

- The entire flash can be updated remotely using the ZFHOST utility, including video BIOS, system BIOS, CMOS setup, and Resident Flash Disk (RFD). There is also a built in capability to copy the information in the flash to your host computer, thereby generating images for manufacturing and field support use. The system BIOS, video BIOS, CMOS setup, and RFD can also be updated separately.
- ZFUTIL utility performs the same tasks as those present in the ZFHOST utility using local (module) resources vs. the host computer resources used with the ZFHOST program except the RFD update. The system BIOS, video BIOS, CMOS setup, and RFD can be updated from a file, uploaded to a file, or erased using the line command of the ZFUTIL utility.
- Automatic detection of ZF board configurations to allow the manipulation of the external devices on the boards with the CMOS CUSTOM configuration screen.
- CMOS setup custom options screen from where additional options can be configured.

Future Release Enhancements

- Advanced power management support. Alpha testing of this feature is currently in process, please contact the factory for additional details and sample code.
- BIOS extensions (user firmware ROM) support in flash

Custom Configuration Screen

This screen allows you to customize the BIOS functionality to your particular application. It eliminates the need to have various flavors of BIOS for the different hardware settings you may need. Some of the items that can be customized are the following (the underlined option is the default):

- System Speed – 100MHz, 66MHz, 33MHz and 8MHz options are available.
- Address 20 latch – Enabled/Disabled
- Console redirection – Probe COM1-4, Always COM1-4, Disabled.
- Redirection Speed – baud rate for redirection (recommend 19.2K baud)
- Manufacturing Mode cable check – Enabled/Disabled
- LPT1 Mode – Bi-direction (PS/2), EPP, Standard (AT), Disabled
- Onboard COM *n* port – Enabled/Disabled
- LPT1 IRQ – Enabled/Disabled
- Super I/O registers – Unlocked/Locked

Release Media

The distribution media containing the Rev E BIOS release consists of 3.5" floppy diskettes. This media contains the following:

ZFUTIL Ver 1.8

ZFUTIL.EXE is a utility for the *OEMmodule 486* and *OEMmodule 486-based* boards. It was designed to be an interactive DOS-environment tool. It supports two modes of operation: a) command line mode for use during re-direction or run from a batch program for automatic upgrades; b) an interactive mode to use if a keyboard and video device are present in the system. It contains on-line help, accessed at the bottom of the ZFUTIL screen giving the user hints, explanation of menu options, etc. This utility contains the functionality of individual *OEMmodule 386* utilities such as FLSHBIOS, RFD-FILE etc. Type *ZFUTIL* for parameter usage details. When you want to load the system or video BIOS, the utility defaults to ZF.ABS (system BIOS) or GENERIC.VBS (video BIOS) you can either rename the appropriate file to use these names or type in the full path and filename at the update prompt.

ZFHOST Ver. 4.4

ZFHOST.EXE is a utility for the *OEMmodule 486* and *OEMmodule 486-based* boards. It was designed to be an interactive DOS-environment tool. In this context HOST is a standalone PC that can be used as a tool that provides a serial connection to the TARGET (device containing the OEMmodule) system. It supports a menu driven mode that uses all the resources of the host PC (video, keyboard, floppy/IDE drives, etc) to perform the tasks shown below. It contains on-line help, accessed at the bottom of the ZFHOST screen giving the user hints, explanation of menu options, etc. This utility contains the functionality previously accomplished using the DOWNLOAD procedure on the 386 module.

The functions present in this utility are the following:

- Get Target Attention
- Target System Information
- Test Target Memory
- Continue loading OS on Target
- Update System BIOS
- Update Video BIOS
- Upload Flash from file (*)
- Download Flash from file (*)
- Erase Flash Block (*)
- Read Target CMOS to File
- Program Target CMOS from File
- Download RFD Image from Target
- Upload RFD Image to Target

- Show Max message size
- Exercise link
- Show Target Drives (*)

(*) not available on this release

Note: These versions of ZFUTIL and ZFHOST have been upgraded to work with rev. E of the 486 BIOS. They are also compatible with earlier revisions of the BIOS (rev. B or later), although some of the functions may not work because rev. E BIOS contains code needed to perform some of the functions.

Setup Files

Tables 1-4 identify the files you will find on the Rev D. release diskettes:

File Name	Description
ZFRN-009.PDF	Release notes. Read this file for important last-minute information.
ZFUTIL.EXE	Performs the functionality found before in the various ZF utilities mentioned below: a) INSTALL.BAT. Batch file. Reformats internal Flash disk, loads new BIOS, and default CMOS configuration. b) UPDATE.BAT. Batch file. Loads the new BIOS, and default CMOS configuration. c) CMOSFILE.EXE. Utility that creates a binary file containing the current CMOS configuration, CMOSFILE.BIN . This file can be used to update the CMOS from a floppy drive. d) FILECMOS.EXE. Utility to copy CMOSFILE.BIN to CMOS memory.
ZFHOST.EXE	A menu driven utility that can be used to update the module flash from a remote computer. The capabilities include downloading the RFD, CMOS, video BIOS, and system BIOS. The extraction of data from the flash can also be performed with this utility.
DR_DOS.EXE	Self-extracting ZIP file with a RFD image with Caldera DR-DOS SYS'ed RFD. Use this RFD image when downloading using ZFHOST.EXE.
CMOS.486	CMOS setup image with the Flash Disk as Drive A: and no Drive B: or Drive C:
ZF486.ABS	Default flavor of the BIOS image.
CRT_04.DAT	Video BIOS for C&T 65545 video chip used to support CRT
H12_03.DAT	Video BIOS for C&T 65545 video chip used to support Hitachi 12.1 panels.
H08_03.DAT	Video BIOS for C&T 65545 video chip used to support Hitachi 8.2 panels.
CRT_04.EXE	Executable file to run CRT video BIOS
H08_03.EXE	Executable file to run Hitachi 8" flat panel video BIOS
H12_03.EXE	Executable file to run Hitachi 12" flat panel video BIOS

Table 1. DR DOS 4.1 Files Diskette

File Name	Description
ZFRN-009.PDF	Release notes. Read this file for important last-minute information.
DR_CRT.IMG	Full RFD image with Caldera DR-DOS SYS'ed RFD, ZF486.ABS BIOS, CRT_04.DAT video BIOS and a CMOS setup with Drive A: as the RFD.
DR_H08.IMG	Full RFD image with Caldera DR-DOS SYS'ed RFD, ZF486.ABS BIOS, H08_03.DAT video BIOS and a CMOS setup with Drive A: as the RFD.
DR_H12.IMG	Full RFD image with Caldera DR-DOS SYS'ed RFD, ZF486.ABS BIOS, H12_03.DAT video BIOS and a CMOS setup with Drive A: as the RFD.
These files are to be used when the fail-safe BIOS ROM (BDIS) jumper is installed (see Installation process, Method B 2.)	

Table 2. DR DOS BDIS Image Diskette

File Name	Description
ZFRN-009.PDF	Release notes. Read this file for important last-minute information.
ZFUTIL.EXE	Performs the functionality found before in the various ZF utilities mentioned below: a) INSTALL.BAT. Batch file. Reformats internal Flash disk, loads new BIOS, and default CMOS configuration. b) UPDATE.BAT. Batch file. Loads the new BIOS, and default CMOS configuration. c) CMOSFILE.EXE. Utility that creates a binary file containing the current CMOS configuration, CMOSFILE.BIN . This file can be used to update the CMOS from a floppy drive. d) FILECMOS.EXE. Utility to copy CMOSFILE.BIN to CMOS memory.
ZFHOST.EXE	A menu driven utility that can be used to update the module flash from a remote computer. The capabilities include downloading the RFD, CMOS, video BIOS, and system BIOS. The extraction of data from the flash can also be performed with this utility.
GS_DOS.EXE	Self-extracting ZIP file with a RFD image with General Software DOS SYS'ed RFD. Use this RFD image when downloading using ZFHOST.EXE.
CMOS.486	CMOS setup image with the Flash Disk as Drive A: and no Drive B: or Drive C:
ZF486.ABS	Default flavor of the BIOS image.
CRT_04.DAT	Video BIOS for C&T 65545 video chip used to support CRT
H12_03.DAT	Video BIOS for C&T 65545 video chip used to support Hitachi 12.1 panels.
H08_03.DAT	Video BIOS for C&T 65545 video chip used to support Hitachi 8.2 panels.
CRT_04.EXE	Executable file to run CRT video BIOS
H08_03.EXE	Executable file to run Hitachi 8" flat panel video BIOS
H12_03.EXE	Executable file to run Hitachi 12" flat panel video BIOS

Table 3. GSI DOS 4.1 Files Diskette

File Name	Description
ZFRN-009.PDF	Release notes. Read this file for important last-minute information.
GSI_CRT.IMG	Full RFD image with General Software DOS SYS'ed RFD, ZF486.ABS BIOS, CRT_04.DAT video BIOS and a CMOS setup with Drive A: as the RFD.
GSI_H08.IMG	Full RFD image with General Software DOS SYS'ed RFD, ZF486.ABS BIOS, H08_03.DAT video BIOS and a CMOS setup with Drive A: as the RFD.
GSI_H12.IMG	Full RFD image with General Software DOS SYS'ed RFD, ZF486.ABS BIOS, H12_03.DAT video BIOS and a CMOS setup with Drive A: as the RFD.
These files are to be used when the fail-safe BIOS ROM (BDIS) jumper is installed (see Installation process, Method B 2.)	

Table 4. GSI DOS BDIS Images Diskette

Flash Architecture

The *OEMmodule 486* has a total of 2Mbytes of internal flash memory that is organized in 32 individual 64Kbyte pages. Flash utilization and starting addresses are documented in Table 1Table 5.

Starting Address	Pages used	Function
80000000h	.5	Video BIOS (32K)
80008000h	.5	Reserved for factory use (32K)
80010000h	1	Reserved for user extension ROM's (64K)
80020000h	.5	Reserved (32K) for factory use
80028000h	.5	CMOS Flash backup (32K, 512bytes used)
80030000h	1	Reserved for external debugger
80040000h	23	RFD
801B0000h	4	Reserved for future enhancements
801F0000h	1	System BIOS

Table 5. Flash Utilization

Installation process

NOTE: It is important that the DiskOnChip firmware update be performed before the BIOS update.

The new BIOS can be installed on your target system using two separate methods. The exact method you choose will be dependent upon the resources you have available in your system.

Method A

If your system has a floppy, IDE or Compact flash device, simply copy the files in this release to the drive you have available. Run the ZFUTIL program with the -i option, if you have a keyboard and monitor attached to your system. Use the arrow keys to navigate to the BIOS menu option and press enter to select the process to be performed from the menu. Run the ZFUTIL program in the line mode command, if you are in re-direction mode. When you want to load the system or video BIOS the utility defaults to ZF.ABS (system BIOS) or GENERIC.VBS (video BIOS) you can either rename the appropriate file to use these names or type in the full path and filename at the update prompt.

Method B

If your system does not have a floppy, IDE or Compact Flash (or equivalent) device, you can use the serial port to download the code. Two different circumstances must be contemplated:

- 1) **If your flash BIOS is at Rev B (12/11/98) or later.** Simply install the download cable on COM1 (or the COM port specified in your custom configuration screen) and run the ZFHOST program on the host computer using the cursor keys to select the MENU item to upgrade the system. Specifications for a Download cable are provided in the OEMmodule 486 Data Book.
- 2) **If your flash BIOS is pre-Rev B or the fail-safe BIOS ROM (BDIS) jumper is installed.** The pre-Rev. B BIOS and the BDIS BIOS does not support the ZFHOST program, therefore, you must install the BDIS jumper on the target system and follow the procedure detailed below.
 - a) With both the host computer and the ZF target system off, connect a download cable between the target ZF system's COM1 serial port and the COM port you will use on the host computer. Specifications for a Download cable are provided in the OEMmodule 486 Data Book.
 - b) Boot the host computer.
 - c) Install the BIOS upgrade diskette in the host computer.
 - d) Copy the files shown in Table 6 to the host computer's hard disk and rename the file you want to use as a Download image as shown in the Download column. (This is required to download them in **Binary** format.)

Setup Disk	Download	Description
DR_CRT.IMG	F-000000	Entire flash image with CRT video and a DR-DOS SYS'ed Flash Disk
DR_H08.IMG	F-000000	Entire flash image with 12 inch Hitachi video and a DR-DOS SYS'ed Flash Disk.
DR_H12.IMG	F-000000	Entire flash image with 8 inch Hitachi video and a DR-DOS SYS'ed Flash Disk.
GSI_CRT.IMG	F-000000	Entire flash image with CRT video and a General Software DOS SYS'ed Flash Disk.
GSI_H08.IMG	F-000000	Entire flash image with 12 inch Hitachi video and a General Software DOS SYS'ed Flash Disk.
GSI_H12.IMG	F-000000	Entire flash image with 8 inch Hitachi video and a General Software DOS SYS'ed Flash Disk.

Table 6. BDIS Image Files

- e) Start the host computer's terminal emulator. You can use any terminal emulator, such as ProComm, that supports **YMODEM Batch** protocol. Be sure the serial port you are using is configured for:
19.2 K baud, No parity, 8 data bits, 1 stop bit
- f) Power up the target ZF system. After a few seconds, it will begin to poll the host computer.
- g) On the host computer, initiate a **YMODEM Batch** transfer of the **F-000000** file you want to download. (If you have a post-code display attached to your system, you can observe the various stages of the download and Flash programming process.)
- h) When the download is complete and the Flash device has been reprogrammed, power-off both systems, remove the serial cable from the ZF system, remove the BDIS jumper, wait ten seconds, and then power it back on. Confirm that the new BIOS is working by observing the revision letter displayed during post.

This completes the download upgrade procedure.

486 Post Code Listing

The following list contains symbol definitions associated with POST progress reporting in chronological, not numerical, order.

POST_STATUS_START	=	00h	; POST beginning.
POST_STATUS_CPUTEST	=	01h	; CPU register test about to start.
POST_STATUS_DELAY	=	02h	; NMIs are disabled; delay starts.
POST_STATUS_DELAYDONE	=	03h	; power-on delay finished.
POST_STATUS_KDBATRDY	=	04h	; kbd BAT done; reading kbd SYS bit.
POST_STATUS_DISABSHADOW	=	05h	; disabling shadowing & cache.
POST_STATUS_CALCKSUM	=	06h	; calcing ROM cksum, wait kbd ctrlr.


```

POST_STATUS_CKSUMGOOD=    07h    ; cksun okay, kbd ctrllr free.
POST_STATUS_BATVRFY      =    08h    ; verifying BAT cmd to kbd ctrllr.
POST_STATUS_KBDCMD       =    09h    ; issuing kbd ctrllr cmd byte.
POST_STATUS_KBDDATA      =    0ah    ; issuing kbd ctrllr data byte.
POST_STATUS_BLKUNBLK     =    0bh    ; issuing pin 23,24 blocking & unblocking.
POST_STATUS_KBDNOP       =    0ch    ; issuing kbd ctrllr NOP cmd next.
POST_STATUS_SHUTTEST     =    0dh    ; testing CMOS RAM shutdown register.
POST_STATUS_CMOSDIAG     =    0eh    ; checking CMOS cksum, updating DIAG byte.
POST_STATUS_CMOSINIT     =    0fh    ; initializing CMOS (if req'd every boot).
POST_STATUS_CMOSSTATUS=   10h    ; init CMOS status reg for date/time.
POST_STATUS_DISABDMAINT =   11h    ; disabling DMA, interrupt ctrllrs.
POST_STATUS_DISABPORTB=  12h    ; disabling Port B, disabling video display.
POST_STATUS_BOARD       =   13h    ; init board, start auto-mem detect.
POST_STATUS_TESTTIMER    =   14h    ; starting timer tests.
POST_STATUS_TESTTIMER2   =   15h    ; testing 8254 T2, for spkr, part B.
POST_STATUS_TESTTIMER1   =   16h    ; testing 8254 T1, for refresh.
POST_STATUS_TESTTIMER0   =   17h    ; testing 8254 T0, for 18.2Hz.
POST_STATUS_MEMREFRESH=  18h    ; starting memory refresh.
POST_STATUS_TESTREFRESH =  19h    ; testing memory refresh.
POST_STATUS_TEST15US     =   1ah    ; testing 15usec refresh ON/OFF time.
POST_STATUS_TEST64KB     =   1bh    ; testing base 64KB memory.
POST_STATUS_TESTDATA     =   1ch    ; testing data lines.
POST_STATUS_TESTADDR     =   20h    ; testing address lines.
POST_STATUS_TESTPARITY   =   21h    ; testing parity (toggling).
POST_STATUS_TESTMEMRDWR =  22h    ; base 64KB mem read/write test.

```

; Now we have memory, so we can use a stack to use Pcall, not Rcall.

```

POST_STATUS_SYSINIT      =   23h    ; system init before vector table init.
POST_STATUS_INITVECTORS =   24h    ; init vector table.
POST_STATUS_8042TURBO    =   25h    ; reading 8042 for turbo switch setting.
POST_STATUS_POSTTURBO    =   26h    ; initializing turbo data.
POST_STATUS_POSTVECTORS =   27h    ; any init after vector table init is next.
POST_STATUS_MONOMODE     =   28h    ; setting monochrome mode.
POST_STATUS_COLORMODE    =   29h    ; setting color mode.
POST_STATUS_TOGGLEPARITY=   2ah    ; toggle parity before optional video ROM test.
POST_STATUS_INITBEFOREVIDEO=2bh    ; init before video ROM check.
POST_STATUS_VIDEOROM     =   2ch    ; control passed to video ROM.
POST_STATUS_POSTVIDEO    =   2dh    ; video ROM returned control.
POST_STATUS_CHECKEGAVGA =   2eh    ; checking for EGA/VGA adapter found.
POST_STATUS_TESTVIDEOMEMORY=2fh    ; no EGA/VGA found, r/w test of video memory.
POST_STATUS_RETRACE      =   30h    ; looking for video retrace signal.
POST_STATUS_ALTDISPLAY   =   31h    ; retrace failed, checking alt. display.
POST_STATUS_ALTRETRACE   =   32h    ; alt found, checking video retrace signal.
POST_STATUS_VRFYSWADAPTER=33h    ; compare switches w/actual adapter type.
POST_STATUS_SETDISPMODE =   34h    ; setting display mode.

```

; Now we have a display. All code that outputs codes at 35h and above
; can use INT 10h to display messages.

```

POST_STATUS_CHECKSEG40A =   35h    ; check ROM BIOS data area at seg 40h.
POST_STATUS_SETCURSOR    =   36h    ; setting cursor for power-on msg.
POST_STATUS_PWRONDISPLAY=   37h    ; displaying power-on message.
POST_STATUS_SAVECURSOR=  38h    ; save cursor position.

```

POST_STATUS_BIOSIDENT	=	39h	; display BIOS ident. string.
POST_STATUS_HITDEL	=	3ah	; display "Hit to ..." msg.
POST_STATUS_VIRTUAL	=	40h	; preparing vm test. vrfy from display memory.
POST_STATUS_DESCR	=	41h	; preparing descriptor tables.
POST_STATUS_ENTERVM	=	42h	; enter virtual mode for memory test.
POST_STATUS_ENABINT	=	43h	; enable ints for diagnostics mode.
POST_STATUS_CHECKWRAP1	=	44h	; init data for checking wraparound at 0:0.
POST_STATUS_CHECKWRAP2	=	45h	; checking for wrap, find total memory size.
POST_STATUS_HIGHPATTERNS	=	46h	; write extended memory test patterns.
POST_STATUS_LOWPATTERNS	=	47h	; write conventional memory test patterns.
POST_STATUS_FINDLOWMEM	=	48h	; finding low memory size from patterns.
POST_STATUS_FINDHIMEM	=	49h	; finding high memory size from patterns.
POST_STATUS_CHECKSEG40B	=	4ah	; check ROM BIOS data area again.
POST_STATUS_CHECKDEL	=	4bh	; check for , clear low mem for soft reset.
POST_STATUS_CLREXTMEM	=	4ch	; clearing ext mem for soft reset.
POST_STATUS_SAVEMEMSIZE	=	4dh	; saving memory size.
POST_STATUS_COLD64TEST	=	4eh	; on cold boot, display 1st 64KB memtest.
POST_STATUS_COLDLOWTEST	=	4fh	; on cold boot, test all of low memory.
POST_STATUS_ADJUSTLOW	=	50h	; adjust memsize for 1K usage.
POST_STATUS_COLDHITEST	=	51h	; on cold boot, test high memory.
POST_STATUS_REALMODETEST	=	52h	; prepare for shutdown to real-mode.
POST_STATUS_ENTERREAL	=	53h	; saved regs & memsize, entering real-mode.
POST_STATUS_SHUTDOWN	=	54h	; shutdown successful, restoring codepath.
POST_STATUS_DISABA20	=	55h	; disabling A20 line.
POST_STATUS_CHECKSEG40C	=	56h	; checking ROM BIOS data area again.
POST_STATUS_CHECKSEG40D	=	57h	; checking ROM BIOS data area some more.
POST_STATUS_CLRHITDEL	=	58h	; clear the "Hit " message.
POST_STATUS_TESTDMAPAGE	=	59h	; test DMA page register.
POST_STATUS_VRFYDISPMEM	=	60h	; verify from display memory (???).
POST_STATUS_TESTDMA0BASE	=	61h	; test DMA0 base register.
POST_STATUS_TESTDMA1BASE	=	62h	; test DMA1 base register.
POST_STATUS_CHECKSEG40E	=	63h	; checking ROM BIOS data area again.
POST_STATUS_CHECKSEG40F	=	64h	; checking ROM BIOS data area some more.
POST_STATUS_PROGDMA	=	65h	; programming DMA ctrlrs 0 & 1.
POST_STATUS_INITINTCTRL	=	66h	; initializing INT ctrlrs 0 & 1.
POST_STATUS_STARTKBDTEST	=	67h	; starting keyboard test.
POST_STATUS_KBDRESET	=	80h	; issuing reset cmd & clearing output buffer.
POST_STATUS_CHECKSTUCKKEYS	=	81h	; check for stuck keys & issue test cmd.
POST_STATUS_INITCIRCBUFFER	=	82h	; initializing circular buffer.
POST_STATUS_CHECKLOCKEDKEYS	=	83h	; check for locked keys.
POST_STATUS_CHECKMEMSIZEMISMATCH	=	84h	; check for memsize mismatch (CMOS/BIOSDATA).
POST_STATUS_PASSWORD	=	85h	; check for pswd or bypass setup.
POST_STATUS_BEFORESETUP	=	86h	; pswd checked. do pgming before setup.
POST_STATUS_CALLSETUP	=	87h	; call the setup module.
POST_STATUS_POSTSETUP	=	88h	; back from setup, clr screen.
POST_STATUS_DISPPWRON	=	89h	; display power-on screen message.
POST_STATUS_DISPWAIT	=	8ah	; display "Wait..." message.
POST_STATUS_ENABSHADOW	=	8bh	; do system & video BIOS shadowing.
POST_STATUS_STDCMOSSETUP	=	8ch	; load standard setup params into BIOSDATA.
POST_STATUS_MOUSE	=	8dh	; check and initialize mouse.
POST_STATUS_FLOPPY	=	8eh	; check floppy disks.
POST_STATUS_CONFIGFLOPPY	=	8fh	; configure floppy drives.
POST_STATUS_IDE	=	90h	; check hard disks.
POST_STATUS_CONFIGIDE	=	91h	; configure IDE drives.

POST_STATUS_CHECKSEG40G = 92h ; checking ROM BIOS data area again.
POST_STATUS_CHECKSEG40H = 93h ; checking ROM BIOS data area some more.
POST_STATUS_SETMEMSIZE = 94h ; setting base & ext mem sizes.
POST_STATUS_SIZEADJUST = 95h ; memsize adjusted for 1K, verifying disp mem.

; Initialize any ROM BIOS extensions.

POST_STATUS_INITBEFOREC8000 = 96h ; initialization before calling C800h.
POST_STATUS_CALLC8000 = 97h ; call ROM BIOS extension at C800h.
POST_STATUS_POSTC8000 = 98h ; processing after extension returns.

; Setup serial ports, parallel ports, NPX, keyboard, cache, wait states.

POST_STATUS_TIMERPRNBASE= 99h ; configuring timer data area, printer base addr.
POST_STATUS_SERIALBASE = 9ah ; configuring serial port base addrs.
POST_STATUS_INITBEFORENPX = 9bh ; initialization before coprocessor test.
POST_STATUS_INITNPX = 9ch ; initializing the coprocessor.
POST_STATUS_POSTNPX = 9dh ; processing after coprocessor initialized.
POST_STATUS_CHECKLOCKS= 9eh ; check ext kbd, kbdID, numlock settings.
POST_STATUS_ISSUEKBDID = 9fh ; issue keyboard ID command next.
POST_STATUS_RESETID = 0a0h ; kbd ID flag reset.
POST_STATUS_TESTCACHE = 0a1h ; do cache memory test.
POST_STATUS_DISPSOFTERR = 0a2h ; display any soft errors.
POST_STATUS_TYPEMATIC = 0a3h ; set keyboard typematic rate.
POST_STATUS_MEMWAIT = 0a4h ; program memory wait states.
POST_STATUS_CLRSCR = 0a5h ; clear screen.
POST_STATUS_ENABPTYNMI = 0a6h ; enable parity and NMIs.

; Initialize ROM BASIC if available.

POST_STATUS_INITBEFOREE000 = 0a7h ; initialization before calling E000h.
POST_STATUS_CALLE000 = 0a8h ; call ROM BIOS extension at E000h.
POST_STATUS_POSTE000 = 0a9h ; processing after extension returns.

; Boot operating system.

POST_STATUS_DISPCONFIG = 0b0h ; display system config box.
POST_STATUS_INT19BOOT = 00h ; call INT 19h bootstrap loader.

; Additional paths.

POST_STATUS_LOWMEMEXH= 0b1h ; test low memory exhaustively.
POST_STATUS_EXTMEMEXH = 0b2h ; test extended memory exhaustively.

OEMmodule™ 486 Version E Release Notes

Revision History:

Revision x **4/8/99**

Revision A **5/6/99**

SMX/386, OEMmodule 386, OEMmodule 486 are trademarks of ZF MicroSystems, Incorporated.
All other trademarks and registered trademarks are the property of their respective owners.



ZF MicroSystems, Incorporated

1052 Elwell Court • Palo Alto, California 94303

Tel: (650) 965-3800 Fax: (650) 965-4050

Home Page: <http://www.zfmicro.com>

Email: support@zfmicro.com

ZF MicroSystems assumes no responsibility or liability for any errors or inaccuracies contained in this document, nor for the use of the information or any portion of this information in any application, including any claim for copyright or patent infringement or direct, indirect, special or consequential damages. There are no warranties extended or granted by this document. The information herein is subject to change without notice from ZF MicroSystems, Incorporated. ©1998 ZF MicroSystems, Incorporated. All rights Reserved.